J Control Theory Appl 2013 11 (4) 563–569
DOI 10.1007/s11768-013-2119-6

Nonlinear model predictive control with relevance vector regression and particle swarm optimization

M. GERMIN NISHA, G. N. PILLAI†

Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee-247667, Uttarakhand, India

Abstract: In this paper, a nonlinear model predictive control strategy which utilizes a probabilistic sparse kernel learning technique called relevance vector regression (RVR) and particle swarm optimization with controllable random exploration velocity (PSO-CREV) is applied to a catalytic continuous stirred tank reactor (CSTR) process. An accurate reliable nonlinear model is first identified by RVR with a radial basis function (RBF) kernel and then the optimization of control sequence is speeded up by PSO-CREV. Additional stochastic behavior in PSO-CREV is omitted for faster convergence of nonlinear optimization. An improved system performance is guaranteed by an accurate sparse predictive model and an efficient and fast optimization algorithm. To compare the performance, model predictive control (MPC) using a deterministic sparse kernel learning technique called Least squares support vector machines (LS-SVM) regression is done on a CSTR. Relevance vector regression shows improved tracking performance with very less computation time which is much essential for real time control.

Keywords: Relevance vector regression; Least squares support vector machines; Nonlinear model predictive control; Particle swarm optimization with controllable random exploration velocity

1 Introduction

Model predictive control (MPC) is recognized as an advanced control technique which has been very successful in practical applications [1]. This acknowledgment is due to its ability to handle constraints imposed on process inputs and outputs, process nonlinearities, dead times, and model uncertainties. In earlier times, linear model predictive controllers were repeatedly used in practice. However, they fail to work with the inevitable nonlinear behavior of chemical processes.

Linear model predictive controllers are inadequate for highly nonlinear processes and moderately nonlinear processes which have large operating regimes. This shortcoming coupled with increasingly stringent demands on throughput and product quality has spurred the development of nonlinear model predictive control [2]. Two challenging tasks in nonlinear model predictive controller are acquiring an accurate nonlinear model and solving nonlinear optimization problem online.

The performance of nonlinear model predictive controller depends on model accuracy. For a highly tuned controller a very accurate model is necessary [3]. Thus, precise nonlinear model is expected for better controlled performance. Neural networks were widely believed for estimation of nonlinear system dynamics due to its simplicity besides its poor extrapolation, poor generalization. Liu et al. [4] have stated that training a neural network is too lengthy and the number of training data required is more. Several scholars [5–7] have approximated nonlinear models by neural networks which show acceptable performance. Despite the existence of many nonlinear control strategies in theory, designing a suitable controller for complex process is still a challenge in practice [4].

The sparse kernel learning is a nonlinear modeling method originally proposed in the machine learning area [8–9]. A deterministic nonlinear modeling method, support vector machines (SVM) which overwhelms the over fitting and poor generalization ability of neural network with less number of training data and less training time providing better tracking performance is introduced in [10]. However, practical applications of SVMs are limited because of its requirement of larger number of kernels to approximate the optimal solutions. In least squares support vector machines (LS-SVM) the regularization parameter γ and the kernel width parameter σ are the two parameters to be tuned to improve the generalization ability of predicted model. Thus, the LS-SVM model is burdened with additional externally determined parameters, which is a time-consuming task. Subsequently, Tipping [11] introduced relevance vector machine (RVM) in 2000 which attracted much interest in the research community owing to its advantages over support vector machine. They are established on a Bayesian formulation which results in using a smaller number of relevance vectors leading to much more sparse representation than support vector machine (SVM). Unlike in SVM framework where the basis functions must satisfy Mercer's kernel theorem, in the RVM case there is no restriction on the basis functions [10, 12]. Also, kernel width σ is the only parameter to be tuned in relevance vector regression (RVR) model. Consequently the sparse RVR model could generalize better with very less computation time than SVM. The result given in [11] demonstrates the comparable generalization performance of RVM than SVM with intensely fewer kernel functions.

Nonlinear system identification using RVM was successfully discussed in many literature [13–21] which highlight

Received 26 May 2012; Revised 25 June 2013.
†Corresponding author. E-mail: gopinath.pillai@gmail.com.

its significance. To our best knowledge, combinations of RVM model and MPC approach has rarely been reported in literature.

Despite of accurate approximation of nonlinear dynamics by RVR model it suffers from computational burden as model predictive controller does prediction and optimization at each sampling instant. Since the particle swarm optimization was developed, owing to its simplicity and high performance, it has been proven to be a powerful competitor to other evolutionary algorithms [22–23], and been widely used in many optimization processes [24–25]. It is a computationally efficient method since it is derivative free.

In this paper, a nonlinear model predictive controller combining relevance vector regression model and particle swarm optimization with controllable random exploration velocity (PSO-CREV) is presented; which merges the advantage of accurate prediction and less computational effort. Use of RVM creates an accurate model for prediction and PSO-CREV parameters make the predictive control faster. Simulation results of a catalytic catalytic continuous stirred tank reactor (CSTR) process illustrates the better tracking performance of RVM-based MPC when compared with LS-SVM-based MPC with much less computation time.

This paper encompasses five sections commencing with the introduction as the first section followed by the second section which describes theory behind relevance vector machine. The third section explains MPC based on RVM and particle swarm optimization. The fourth section shows a comparative study of a CSTR with RVM based MPC and LS-SVM-based MPC controller with suitable simulation results and the fifth section concludes the paper.

2 Relevance vector machine

RVM is a probabilistic model whose functional form is equivalent to that of SVM. It achieves comparable recognition accuracy to the SVM, yet provides a full predictive distribution, and also requires substantially fewer kernel functions [26]. RVM is based on Bayesians approach in which a prior is introduced over the model weights and each weight is administrated by one hyperparameter. The most probable value of each hyper parameter is iteratively evaluated from the data. The model is sparser since the posterior distributions of some proportion of the weights are set to zero.

Consider a given training set of M regression data points $\{(x_m, y_m)\}_{m=1}^{M}$, where $x_m \in \mathbb{R}^M$ is the input data to the actual plant and $y_m \in \mathbb{R}$ is the output data of the actual plant and is assumed to contain Gaussian noise ε with mean 0 and variance σ^2. In high dimensional feature space z, the outputs of an extended linear model can be expressed as a linear combination of the response of a set of M basis functions as follows:

$$y(x, w) = \sum_{m=1}^{M} w_m \varphi_m(x) + \varepsilon = w^{\mathrm{T}}\varphi + \varepsilon. \quad (1)$$

Now, the predicted output $\hat{y}$ of the true value y is

$$\hat{y}(x, w) = \sum_{m=1}^{M} w_m \varphi_m(x) = w^{\mathrm{T}}\varphi \ \text{ where } \ w \in z. \quad (2)$$

In the above nonlinear function estimation model, w_m is the weight vector and $\varphi_m(\cdot)$ is an arbitrary basis function (or kernel). In the present work, RBF is used as the kernel function because of its ability to reduce computational complexity of the training process. The vector form of $w = [w_1 \ \cdots \ w_M]^{\mathrm{T}}$ and the responses of all kernel function $\varphi(x) = [\varphi_1(x) \ \cdots \ \varphi_M(x)]^{\mathrm{T}}$ maps the input data into a high dimensional feature space z.

Hence, the obtained error signal could be stated as

$$\varepsilon_m = y_m - \hat{y}_m = N(0, \sigma^2). \quad (3)$$

The objective of relevance vector regression is to find the finest value of w such that $\hat{y}(x, w)$ makes good predictions for unknown input data. For the RVM model in equation (2) let $\alpha = [\alpha_1 \ \cdots \ \alpha_M]^{\mathrm{T}}$ be the vector of M independent hyperparameters, each associated with one model weight or kernel function.

The Gaussian prior distributions of the RVM framework are chosen as

$$p(\frac{w_m}{\alpha_m}) = \prod_{m=1}^{M} (\frac{\alpha_m}{2\pi})^{\frac{1}{2}} \exp\{-\frac{\alpha_m w_m^2}{2}\}. \quad (4)$$

Here, α_m is the hyperparameter that governs each weight w_m. The likelihood function of independent training targets $y = y_m, m = 1, \ldots, M$ can be stated as

$$p(\frac{y}{w}, \sigma^2) = \prod_{m=1}^{M} p(\frac{y_m}{w}, \sigma^2) = \frac{\mathrm{e}^{-\frac{\|(y-\hat{y})^2\|}{2}\sigma^2}}{\sqrt{(2\pi\sigma^2)^M}}. \quad (5)$$

The above likelihood function is enhanced by the prior in equation (4) defined over each weight to reduce the complexity of the model and to avoid over fitting. Now, using Bayes' rule, the posterior distribution over model weights could be calculated as follows:

$$p(\frac{w}{y}, \alpha, \sigma^2) = \frac{p(\frac{y}{w}, \sigma^2) p(\frac{w}{\alpha})}{p(\frac{y}{\alpha}, \sigma^2)}. \quad (6)$$

The posterior distribution in equation (6) is a Gaussian distribution function,

$$p(\frac{w}{y}, \alpha, \sigma^2) = N(\mu, \sigma^2), \quad (7)$$

whose covariance and mean are respectively given by

$$\Sigma = (\sigma^{-2}\varphi^{\mathrm{T}}\varphi + A)^{-1}, \quad (8)$$

$$\mu = \sigma^{-2}\Sigma\varphi^{\mathrm{T}}y \quad (9)$$

with $A = \mathrm{diag}\{\alpha\}$.

Marginalization of the likelihood distribution over the training targets given by equation (5) can be obtained by integrating out the weights to acquire the marginal likelihood for the hyperparameters.

$$p(\frac{y}{\alpha}, \sigma^2) = \int p(\frac{y}{w}, \sigma^2) p(\frac{w}{\alpha}) \mathrm{d}w = N(0, C). \quad (10)$$

Here, the covariance is given by $C = \sigma^2 I + \varphi A^{-1}\varphi^{\mathrm{T}}$. In equations (8) and (9), the only unknown variables are the hyperparameters α. The values of these hyperparameters are estimated using the framework of type II maximum likelihood [27].

$$\log p(\frac{y}{\alpha}, \sigma^2) = -\frac{1}{2}(M \log 2\pi + \log|C| + y^{\mathrm{T}}C^{-1}y). \quad (11)$$

Logarithm is included in equation (11) to reduce computational complexity. Maximization of the logarithmic marginal likelihood in equation (11) over α leads to the most probable value α_{MP} which provides the maximum a

posteriori (MAP) estimate of the weights.

The ambiguity about the optimal value of the weights, given by (6), is used to express ambiguity about the predictions made by the model, i.e., given an input x^*, the probability distribution of the corresponding output y^* is given by the predictive distribution

$$p(\frac{y^*}{x^*}, \hat{\alpha}, \hat{\sigma}^2) = \int p(\frac{y^*}{x^*}, w, \hat{\sigma}^2) p(\frac{w}{y}, \hat{\alpha}, \hat{\sigma}^2) \mathrm{d}w, \quad (12)$$

which has the Gaussian form

$$p(\frac{y^*}{x^*}, \hat{\alpha}, \hat{\sigma}^2) = N(Y^*, \sigma^{*2}). \quad (13)$$

The mean and variance of the predicted model are, respectively,

$$Y^* = \varphi^{\mathrm{T}}(x^*)\mu \text{ and } \sigma^{*2} = \hat{\sigma}^2 + \varphi^{\mathrm{T}}(x^*)\Sigma\varphi(x^*). \quad (14)$$

Maximizing the logarithmic marginal likelihood in (11) leads the optimal values of many of the hyperparameters α_m typically infnite yielding a posterior distribution in (6) of the corresponding weights w_m that tends to be a delta function peaked to zero. Thus, the corresponding weights are deleted from the model along with its accompanying kernel function. Hence, very few data points corresponding to nonzero weights build the RVM model and are called the relevance vectors. This results in better sparseness of RVM model than SVM model. Thus, the computation time for prediction using RVM model is reduced significantly.

3 MPC based on RVM and particle swarm optimization

3.1 RVM-based MPC principle

The basic structure of RVM-based nonlinear model predictive controller is shown in Fig. 1. It includes three important blocks, the actual plant to be controlled with output $y(k) = [y_1(k) \ \cdots \ y_M(k)]^{\mathrm{T}}$. Then, the RVM model of the actual plant to be controlled with predicted output $\hat{y}(k) = [\hat{y}(k+1) \ \cdots \ \hat{y}(k+N_{\mathrm{p}})]$ here. N_{p} is the prediction horizon of MPC which dictates how far we wish the future to be predicted for. Next is the optimization block which provides the optimized control signal $u(k) = [u_1(k) \ \cdots \ u_{N_{\mathrm{u}}}(k)]$, where N_{u} is the control horizon of MPC which dictates the number of control moves used to attain the future control trajectory, subjected to the specified constraints which are required for the plant to achieve the desired trajectory $\mathrm{ref}(k) = [\mathrm{ref}(k+1) \ \cdots \ \mathrm{ref}(k+N_{\mathrm{p}})]$. Here, k stands for the current sampling instant.

Thus, at each sampling instant a sequence of manipulated variable $u(k)$ is calculated in order to minimize the formulated performance index in (16), i.e., the difference between the predicted output of the model and the desired reference trajectory over the specified prediction horizon N_{p}. The number of manipulated variables in the sequence is decided by the control horizon value N_{u} and only the first manipulated variable in the control sequence is applied to the actual plant. This course is repeated at each sampling instant. The basic structure of LS-SVM-based nonlinear model predictive control is obtained by replacing RVM model with LS-SVM model in Fig. 1.

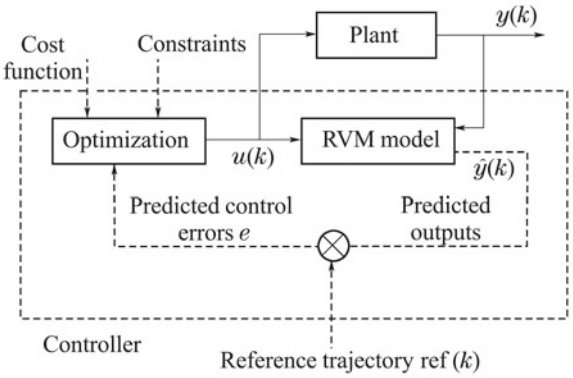

Fig. 1 Basic structure of RVM-based nonlinear model predictive control.

3.2 Performance index formulation

For a single input single output (SISO) nonlinear process the predicted output of RVM model is a function of the past process outputs, $Y(k) = [y(k) \ \cdots \ y(k-n_y+1)]$ and past process inputs, $U(k-1) = [u(k-1) \ \cdots \ u(k-n_u+1)]$. The number of the past controlled outputs and past manipulated inputs depends on the corresponding process orders n_u and n_y, respectively.

Thus, a single step ahead prediction of a SISO nonlinear process can be illustrated by the following discrete time model:

$$\hat{y}(k+1) = f(y(k), \ldots, y(k-n_y+1), u(k), \ldots, u(k-n_u+1)), \quad (15)$$

where k is the discrete time index.

The above equation can be rewritten as

$$\hat{y}(k+1) = f(Y(k), u(k), U(k-1)). \quad (16)$$

Here, $Y(k)$ and $U(k-1)$ are the vectors holding the past controlled outputs and past manipulated inputs, respectively. Thus, after system identification using the regression data set the one step ahead predicted output of RVR model could be formulated as

$$\hat{y}(k+1) = \sum_{m=0}^{M} w_m \varphi_m(x) = w^{\mathrm{T}}\varphi, \quad (17)$$

where M is the number of subsets of training samples.

Accordingly, the performance index to be minimized to achieve the optimal control sequence $u(k)$ can be obtained as

$$J[u(k)] = \sum_{j=N_1}^{N_2} [\mathrm{ref}(k+j) - \hat{y}(k+j)]^2 + \sum_{j=1}^{N_{\mathrm{u}}} \lambda[\Delta u(n+j)]^2. \quad (18)$$

where

N_1	minimum prediction horizon,
N_2	maximum prediction horizon,
N_{u}	control horizon,
$\mathrm{ref}(\cdot)$	reference trajectory,
$\hat{y}(\cdot)$	predicted output of LS-SVM,
Δu	control input change defined as $u(k+j) - u(k+j-1)$,
k	current sampling instant, and
λ	control input weighting factor.

In the performance index formulated in equation (18), $\hat{y}$ depends on the kernel function which in turn is a function of manipulated variable u, which is optimized and applied to the actual plant in order to minimize the deviation between the reference value and controlled variable.

3.3 Conventional particle swarm optimization

Although nonlinear predictive controller is good at controlling unknown nonlinear systems, it does not mean that practical implementation is without difficulties. The primary shortage results from its computational cost [28–29]. Usage of evolutionary algorithm for MPC optimization overcomes this difficulty.

Inspired by the foraging behavior of birds, American psychologist Kennedy and electrical engineer Eberhart developed the particle swarm optimization algorithm [23]. This evolutionary algorithm has the capability of universality and global optimization.

If in an n-dimensional search space, the swarm $X = [X_1 \ \ X_2 \ \cdots \ X_m]$ is composed of m particles. Let the position and velocity of ith individual particles be $X_i = [x_{i1} \ \ x_{i2} \ \cdots \ x_{in}]^{\mathrm{T}}$ and $V_i = [v_{i1} \ \ v_{i2} \ \cdots \ v_{in}]^{\mathrm{T}}$, respectively, and the best position be $P_i = [P_{i1} \ \ P_{i2} \ \cdots \ P_{in}]^{\mathrm{T}}$. Let the global best position be $P_{\mathrm{g}} = [p_{\mathrm{g}1} \ \ p_{\mathrm{g}2} \ \cdots \ p_{\mathrm{g}n}]^{\mathrm{T}}$. Then, the updated velocity and position of particle X_i will be as in equations (19) and (20).

$$v_{id}^{(t+1)} = \omega v_{id}^{(t)} + C_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{\mathrm{g}d}^{(t)} - X_{id}^{(t)}), \tag{19}$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)}, \tag{20}$$

where $d = 1, 2, \ldots, n, i = 1, 2, \ldots, m$, and

m	swarm size,
t	iteration counter,
ω	inertia weights,
r_1, r_2	random numbers in the range [0,1], and
c_1, c_2	learning factors.

Learning factors c_1 and c_2 usually equals to 2. However, other settings were also used. Usually, c_1 equals to c_2 and ranges from [0, 4].

3.4 Disadvantages of conventional PSO

From equations (19) and (20), it is understood that the strength of exploration performance is merely determined by the degrading rate of $(P_{id}^{(t)} - X_{id}^{(t)})$ and $(P_{\mathrm{g}d}^{(t)} - X_{id}^{(t)})$ as r_1 and r_2 are supplemented as relational coefficients to $(P_{id}^{(t)} - X_{id}^{(t)})$ and $(P_{\mathrm{g}d}^{(t)} - X_{id}^{(t)})$, respectively. Hence, if a swarm converges to a local minimal solution, the algorithm may not have the capability to neglect it and hence the strength of exploration behavior of the conventional PSO algorithm needs improvement. This task of improving the exploration strength is achieved in a modified novel algorithm PSO-CREV.

3.5 PSO-CREV algorithm

The intensity of exploration capability of conventional PSO was improved significantly by Chen et al. [28–29], after incorporating some modifications in the position and velocity equations as shown in equations (21) and (22), respectively.

$$v_{id}^{(t+1)} = \varepsilon^{(t)} [\omega v_{id}^{(t)} + c_1 r_1 (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 r_2 (P_{\mathrm{g}d}^{(t)} - X_{id}^{(t)}) + \xi_{id}^{(t)}], \tag{21}$$

$$x_{id}^{(t+1)} = \alpha x_{id}^{(t)} + v_{id}^{(t+1)} + \frac{1-\alpha}{\phi_{id}^{(t)}} [c_1 r_1 P_{id}^{(t)} + c_2 r_2 P_{\mathrm{g}d}^{(t)}], \tag{22}$$

where $\xi_{id}^{(t)}$ is a bounded random variable with continuous uniform distribution, $\varepsilon^{(t)}$ tends to zero as t increases, and $\sum_{t=1}^{\infty} \varepsilon(n) = \infty$, and α ranges between 0 and 1.

The introduction of the terms $\varepsilon(n)$, ξ and the positive coefficient α makes the algorithm different from the conventional PSO. $\varepsilon(n)$ is different from inertia constant as it is operating on all the three components of velocity update equation. A non-zero value of $\xi(n)$ is very useful to drive particles into unknown solution space besides the effect brought by cognitive and social components of the PSO algorithm. Without additional stochastic behavior or $\xi(n) = 0$, PSO-CREV behaves much like the conventional PSO, but with relatively faster convergence. Hence, in this paper this additional stochastic behavior is omitted to achieve faster convergence. A decreasing $\xi(n)$ and a positive coefficient α guarantee faster convergence of the algorithm [29]. Other PSO-CREV parameters are

$$\varepsilon(n) = \frac{3.5}{(n+1)^{0.4}},$$
$$c_1 = c_2 = 2 \text{ and } \alpha = 0.95.$$

4 Application on catalytic CSTR process

This section describes the improved performance of RVM-based nonlinear MPC than LS-SVM-based nonlinear MPC. A CSTR is the chemical process used for illustration. The physical arrangement of CSTR process is shown in Fig. 2. In a catalytic reactor, the rate of catalytic reaction is proportional to the amount of catalyst the reagents contact.

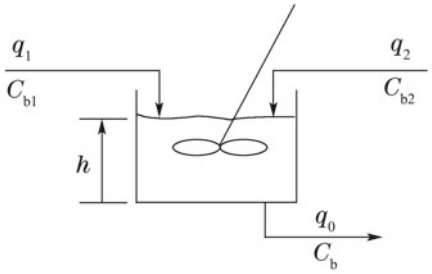

Fig. 2 Schematic of the CSTR process.

The dynamic model of the catalytic CSTR process is

$$\frac{\mathrm{d}h(t)}{\mathrm{d}t} = q_1(t) + q_2(t) - 0.2\sqrt{h(t)}, \tag{23}$$

$$\frac{dC_{\mathrm{b}}(t)}{\mathrm{d}t} = (C_{\mathrm{b}1} - C_{\mathrm{b}}(t))\frac{q_1(t)}{h(t)} + (C_{\mathrm{b}2} - C_{\mathrm{b}}(t))\frac{q_2(t)}{h(t)} - \frac{k_1 C_{\mathrm{b}}(t)}{(1 + k_2 C_{\mathrm{b}}(t))^2}, \tag{24}$$

where

$h(t)$ liquid level in the reactor,
$C_{\mathrm{b}}(t)$ product concentration at the output of the process,
$q_1(t)$ flow rate of the concentrated feed C_{b1},
$q_2(t)$ flow rate of the diluted feed C_{b2}, and
q_0 product flow rate at the output of the process.

The nominal conditions for the feed concentrations are set to $C_{\mathrm{b2}} = 24.9\,\mathrm{mol}\cdot\mathrm{L}^{-1}$ and $C_{\mathrm{b1}} = 0.1\,\mathrm{mol}\cdot\mathrm{L}^{-1}$. The rate of consumption of both the feeds are associated with the constants $k_1 = 1$ and $k_2 = 1$. The objective of this catalytic reactor is to obtain the desired concentration C_{b} at the product by adjusting the feed flow rate q_1 and q_2. The illustration is made simple by setting $q_2(t) = 0.1\,\mathrm{L}\cdot\mathrm{min}^{-1}$.

4.1 Training and testing the model

A sequence of 300 samples is used to train the sparse Bayesian RVR model offline. Hyper parameter estimation is carried out by expectation maximization (EM) updates on the objective function [12]. For this RVR model RBF kernel is used with the width parameter estimated automatically by the learning procedure [12] which improves generalization ability and reduces computational complexity of the training process. Thus, unlike in LS-SVM there is no necessity for computationally expensive determination of regularization parameter by cross validation technique. Also in the RVR model confidence intervals, likelihood values and posterior probabilities could be explicitly encoded easily.

The SVR model is also trained offline using a sequence of 300 samples using the leave one out method. Leave one out method is one in which at each iteration, one leaves out one point and fits the model on the other data points. The performance of the model is estimated based on the point left out. This procedure is repeated for each data point. Finally, all the different estimates of the performance are combined (default by computing the mean). The assumption is made that the input data is distributed independently and identically over the input space [30].

The minimum prediction horizon N_1, maximum prediction horizon N_2 and control horizon N_{u} of RVM-based MPC and LS-SVM-based MPC are set to 1, 1 and 1, respectively to reduce the computational burden. The control input weighting factor λ is set to 0.5.

Figs. 3 and 4 correspond to the modeling results of RVR model and SVR model, respectively. The datasets are generated by providing random constrained signal as input to the plant. The constraint to the input signal is $0 \leqslant u(t) \leqslant 4$. The training error of RVR model and SVR model are shown in Fig. 3, which proves the better training accuracy of RVR model when compared to SVR model.

The trained RVR model and SVR model are further tested with 100 samples of random inputs which are beyond the training data and there corresponding absolute prediction errors are calculated. The comparative graph of prediction errors of RVR model and SVR model are revealed in Fig. 4, which explores a little better extrapolation capability of RVR model than SVR model. Thus, one can conclude that the RVR-based empirical model can prevail little better over the extrapolation capability of SVR model.

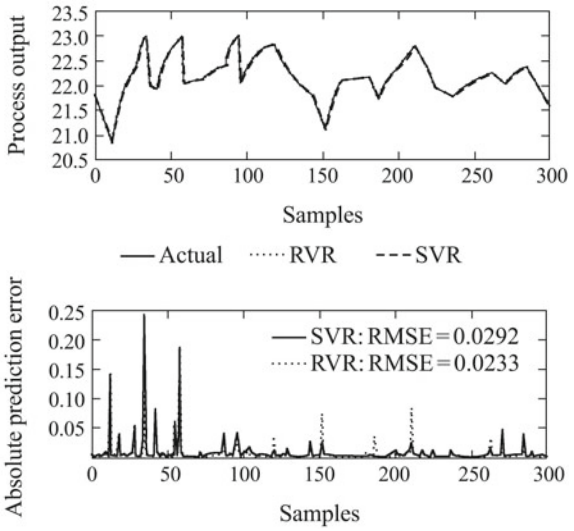

Fig. 3 Training performance comparison of RVR and SVR.

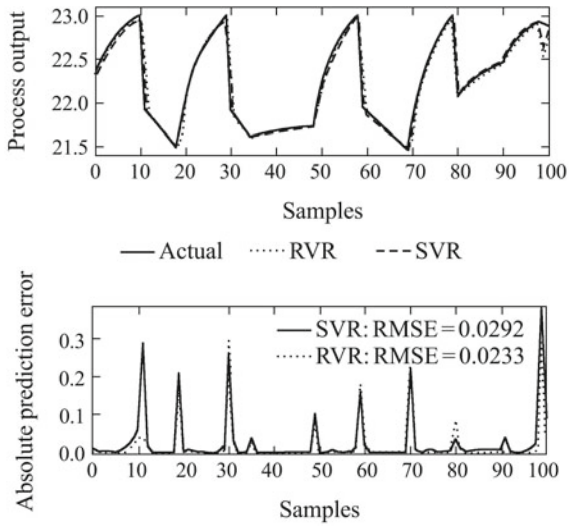

Fig. 4 Testing performance comparison of RVR and SVR models.

The offline trained and validated RVR model or SVR model is then used as the nonlinear model for nonlinear MPC. Fig. 5 illustrates the random set point tracking performances of RVR model-based MPC with PSO-CREV, and SVR model-based MPC with PSO-CREV. Fig. 6 shows the corresponding changes in the process variable, flow rate.

Certainly the tracking performance of RVR-based MPC is little improved when compared with SVR-based MPC's. Both RVR-PSO-CREV and LS-SVM-PSO-CREV-based MPC's are free from overshoots and undershoots due to their accurate predictions and precise optimization using PSO-CREV. However, RVR-PSO-CREV rises little faster than SVR-PSO-CREV which is due to little better generalization and extrapolation capability of accurate RVR model. As the PSO-CREV algorithm converges to the best solution at each sampling instant, the control variable flow rate corresponding to RVM-PSO-CREV and LS-SVM-PSO-CREV are with very less fluctuations.

The control variable $u(k)$, i.e., flow rate of the concentrated feed C_{b1}, is shown in Fig. 6, whose smoothness shows the index of control performance.

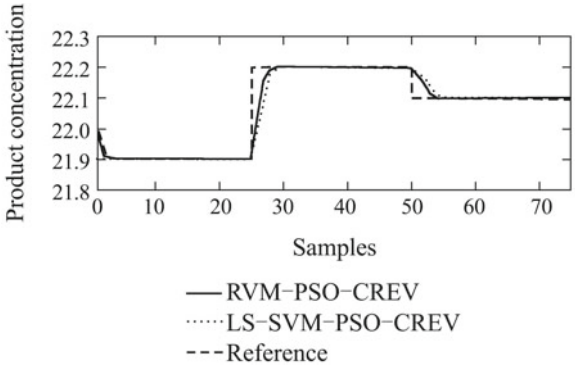

Fig. 5 Tracking performance comparison of product concentration for CSTR process.

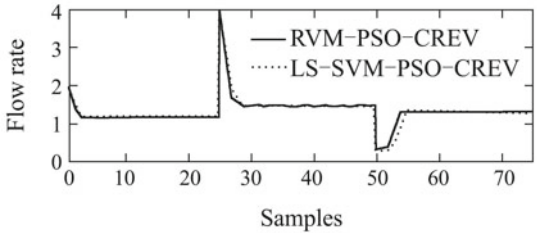

Fig. 6 Changes in the process variable for tracking the product concentration of CSTR process.

The unmeasured disturbance rejection capability of RVM-PSO-CREV-based MPC and LS-SVM-PSO-CREV-based MPC are compared by subjecting the CSTR process with dissimilar magnitudes of disturbance at different sampling instants as shown in Figs. 7 and 8. The control variable, flow rate of the concentrated feed C_{b1} with disturbances at different sampling instance are shown in Fig. 7. Certainly the unmeasured disturbance rejection performance of RVM-PSO-CREV-based MPC is better when compared to LS-SVM-PSO-CREV-based MPC. Thus, RVM-PSO-CREV-based MPC behaves suitably for process control industrial applications.

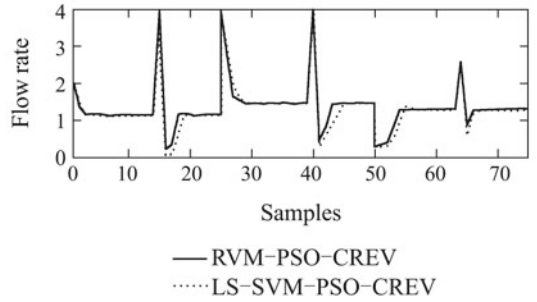

Fig. 7 Changes in the process variable to show unmeasured disturbance.

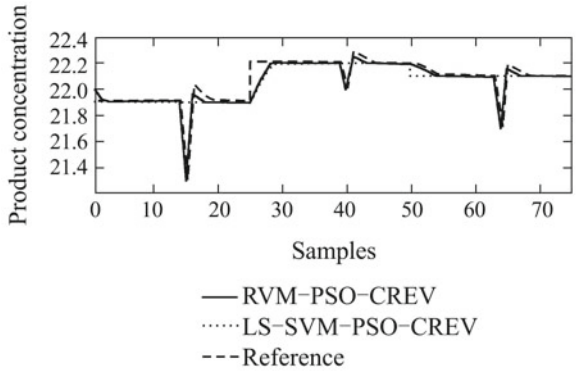

Fig. 8 Performance comparison of unmeasured disturbance rejection.

4.2 Tabulation of performance indices for different controlling techniques

This section enunciates the performance indices and computational cost of both the controllers discussed in previous section. Table 1 shows the integral absolute error (IAE) value, number of support vectors and computational time related to each controller for the simulation results carried out for 75 samples.

IAE is the performance criteria which quantifies the accuracy of both controllers. The number of relevance vectors of RVR model is very less than the number of support vectors of LS-SVM model which sharply reduces the computational time of RVR-MPC to 4.71 seconds for 75 samples (i.e., nearly 0.06 for sample), which is much shorter than the sampling time of the catalytic CSTR process(i.e., 0.2 seconds) under simulation.

Thus, from the above tabulation it is clear that SVR-PSO-CREV model predictive controller is the one which consumes extra time with more IAE and RVR-PSO-CREV model predictive controller is the best controller with very less computational load and little less IAE which is more important for real time applications.

Thus, when compared to LS-SVM-PSO-CREV-based MPC, RVM-PSO-CREV-based MPC is the best controller based on various attributes like little better prediction accuracy, better generalization capability, little better set point tracking performance, better unmeasured disturbance rejection capability with very less computation time due to its sparse model. Hence, it is well suitable for industrial process control applications.

Table 1 Performance indices of various control strategies.

Conditions	Control tactics	Number of training samples	IAE	Number of support vector/ relevance vectors	Computational time/s
No disturbance	LS-SVM-PSO-CREV	300	0.7619	162	10.06
	RVM-PSO-CREV	300	0.6640	27	4.71
Disturbance	LS-SVM-PSO-CREV	300	2.8838	162	11.40
	RVM-PSO-CREV	300	2.2933	27	4.78

5 Conclusions

A viable solution to the problem of fast implementation of nonlinear model predictive control is proposed in this paper. A probabilistic sparse kernel learning technique, RVM is used to create an accurate for prediction model and a derivative free optimization method, PSO-CREV is used to achieve faster convergence. Based on the simulation results of CSTR process the tracking performance of RVM- PSO-CREV-based MPC is superior to LS-SVM-PSO-CREV-based MPC with very less computational cost and better unmeasured disturbance rejection capability which confirms its feasibility. Simulation results convey that such better performance is due to better prediction accuracy, more sparseness property of RVM model and fast accurate convergence of PSO-CREV algorithm.

References

[1] S. Qin, A. T. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 2003, 11(7): 733 – 764.

[2] M. A. Henson. Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, 1998, 23(2): 187 – 202.

[3] J. A. Rossiter. *Model Based Predictive Control: A Practical Approach*. New York: CRC press, 2003.

[4] Y. Liu, Y. Gao, Z. Gao, et al. Simple nonlinear predictive control strategy for chemical processes using sparse kernel learning with polynomial form. *Industrial & Engineering Chemistry Research*, 2010, 49(17): 8209 – 8218.

[5] N. Bhat, T. J. Mcavoy. Use of neural nets for dynamic modeling and control of chemical process systems. *Computers and Chemical Engineering*, 1990, 14(5): 573 – 583.

[6] D. C. Psichogios, L. H. Ungar. Direct and indirect model based control using artificial neural network. *Industrial & Engineering Chemistry Research*, 1991, 30(12): 2564 – 2573.

[7] K. J. Hunt, K. Sbarbaro, R. Zbikowski, et al. Neural network for control systems – a survey. *Automatica*, 1992, 28(6): 1083 – 1120.

[8] J. S. Taylor, N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge: Cambridge University Press, 2004.

[9] C. M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.

[10] V. Vapnik. *Statistical Learning Theory*. New York: Wiley, 1998.

[11] M. E. Tipping. The relevance vector machine. *Advances in Neural Information Processing Systems*. Cambridge: MIT Press, 2000: 652 – 658.

[12] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 2001, 1(3): 211 – 244.

[13] Y. Liu, W. Wang. Sparse Kernel Learning based nonlinear predictive controllers. *Applied mathematics and information sciences*, 2011, 5(2): 195 – 201.

[14] G. Camps-Valls, M. Martinez-Ramon, J. L. Rojo-Alvarez, et al. Nonlinear system identifiation with composite relevance vector machines. *IEEE Signal Processing Letters*, 2007, 14(4) : 279 – 282.

[15] I. Psorakis, T. Damoulas, M. A. Girolami. Multiclass relevance vector machines: sparsity and accuracy. *IEEE Transactions on Neural Networks*, 2010, 21(10): 1588 – 1598.

[16] P. Samui, V. R. Mandla, A. Krishna, et al. Prediction of rainfall using support vector machine and relevance vector machine. *Earth Science India*, 2011, 4(4): 188 – 200.

[17] J. Q. Candela, L. K. Hansen. Time serier prediction based on the relevance vector machine with adaptive kernels. *Proceedings of IEEE International Conference on Aquostics, Speech and Signal Processing*. New York: IEEE, 2003: 985 – 988.

[18] P. Samui, P. Kurup. Use of relevance vector machine for prediction of overconsolidation ratio. *International Journal of Geomechanics*, 2011, 13(1): 26 – 32.

[19] M. A. Nicolaou, H. Gunes, M. Pantic. Output-associative RVM regression for dimensional and continuous emotion prediction. *Image and Vision Computing*, 2012, 30(3): 186 – 196.

[20] S. Tai. An annealing dynamical learning-based relevance vector regression algorithm for housing price forecasting. *Journal of Information and Computational Science*, 2011, 8(14): 3313 – 3319.

[21] P. K. Wong, Q. Xu, C. M. Vong, et al. Rate-dependent hysteresis modeling and control of a Piezostage using online support vector machine and relevance vector machine. *IEEE Transactions on Industrial Electronics*, 2012, 59(4): 1988 – 2001.

[22] R. C. Eberhart, J. Kennedy. A new optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science*. New York: IEEE, 1995: 39 – 43.

[23] J. Kennedy, R. C. Eberhart. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*. New York: IEEE, 1995: 1942 – 1948.

[24] H. Yoshida, K. Kawata, Y. Fukuyama, et al. A particle swarm optimization for reactive power and voltage control considering voltage stability. *Proceedings of IEEE International Conference on Intelligent System Applications to Power Systems*. Rio de Janeiro, 1999: 117 – 121.

[25] L. Messerschmidt, A. P. Engelbrecht. Learning to play games using a PSO-based competitive learning approach. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 280 – 288.

[26] C. M. Bishop, M. E. Tipping. Variational relevance vector machines. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers Inc., 2000: 46 – 53.

[27] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. New York: Springer-Verlag, 1980.

[28] X. Chen, Y. Li. Neural network predictive control for mobile robot Using PSO with controllable random exploration velocity. *International Journal of Intelligent Control and Systems*, 2007, 12(3): 217 – 229.

[29] X. Chen, Y. Li. A modified PSO structure resulting in high exploration ability with convergence guaranteed. *IEEE Transactions on Systems, Man and Cybernetics*, 2007, 37(5): 1271 – 1289.

[30] K. D. Brabanter, P. Karsmakers, F. Ojeda, et. al. *LS-SVM Lab Toolbox User's Guide (Version 1.8)*. Belgium: Heverlee, 1980.

M. GERMIN NISHA received her M.E degree in Process Control and Instrumentation Engineering from Annamalai University in 2005, and currently she is pursuing her Ph.D. at Electrical Engineering Department, Indian Institute of Technology (IIT), Roorkee. Her research interests are in the area of control and artificial intelligence. E-mail: germinnisha@gmail.com.

G. N. PILLAI received his M.Tech. degree from Regional Engineering College, Kurukshetra, India, in 1989, and Ph.D. degree from Indian Institute of Technology (IIT), Kanpur, India, in December 2001. He did his postdoctoral studies from Ulster University, U.K., in 2003. Currently, he is an associate professor at Electrical Engineering Department, Indian Institute of Technology (IIT), Roorkee, India since 2004. Before joining IIT Roorkee, he was a faculty member of Electrical Engineering Department, NIT Kurukshetra, India. His interests are in the areas of power systems, control and artificial intelligence. E-mail: gopinath.pillai@gmail.com.